HW#10 (Group): Due by 11.30pm on 11-01-2020

Instructions

- Save your code to either a **.py** or.**ipynb** files as follows:
  - Either have all of the code in one file named **group<#>_hw10.py** or **group<#>_hw10.ipynb**. In this case, your file should contain an entry labeled **q<#>** with input/output requirements specified by the question.
  - Alternatively, submit all of the questions separately and name the submission files as **group<#>_hw10_q<#>.py** or **group<#>_hw10_q<#>.ipynb**
- Notes:
  - Your code may call other functions that you create, but they have to be included in the file *(i.e., the file that you submit should be self-contained)*
  - If your code calls on functions from other libraries, you need to load them within the function *(e.g., if you use the os library you should assume that it has been installed on the computer but it has not been imported before calling your function)*

Questions

q1 Gini coefficient is a measure of inequality that compares the perfect equality and the actual distribution. Specifically, Gini coefficient is obtained by using the Lorenz curve as follows: calculate the ratio between the area under the Lorenz Curve and the area under the line of equality. [Hint: see here for an example]. Using the data calculate and plot the timeseries of Gini coefficient from 1967 until 2019. Based on your results, what can you say about the income inequality in the US in the last 50 years?

q2 Consider a seller who is selling a perishable item (e.g., fresh lettuce) at price P. Suppose that daily demand for this item can take on any value $D \in [d_{min}, d_{max}]$ with each value equally likely. That is D is uniformly distributed on the interval $[d_{min}, d_{max}]$. Also, suppose that the seller orders the item the night before at cost C. Assume that the item is delivered first thing in the morning, and all unsold items go bad overnight. Write a function that finds the number of units x that the seller should order to maximize his

expected daily profit ep. Your function should return x and ep. Your function should be general so that I can provide my own d_{min}, d_{max}, P, and C when testing.

def findBestQuantity(Dmin,Dmax, P, C):

…

return x, ep

Hints: Profit is revenue minus cost. Revenue is the number of units sold times price, P. Cost is the number of units ordered times cost per unit, C. Notice that the seller can only sell what he has. This means that if the seller orders x and the realized demand is $D > x$ then the seller will sell x units, with revenue $x \times P$ and cost $x \times C$.

- notice that because D is uniformly distributed, the probability that $D > x$ is

$$\Pr(D > x) = \frac{(d_{max} - x)}{(d_{max} - d_{min})} \quad \textit{(assuming that } x \in [d_{min}, d_{max}]\textit{)}$$

If the seller orders x and the realized demand is $d_i \leq x$ then the seller will sell d_i units, with revenue $d_i \times P$ and cost $x \times C$

- notice that because D is uniformly distributed, the probability that $D \leq x$ is

$$\Pr(D \leq x) = \frac{(x - d_{min})}{(d_{max} - d_{min})} \quad \textit{(assuming that } x \in [d_{min}, d_{max}]\textit{)}$$